

Προγραμματιστικές Τεχνικές - Πτυχιακή 2021 (25/2/2021)

ηλεκτρονικά στο MS Forms από απόσταση - διάρκεια 1:30

Το προγραμματιστικό θέμα μετράει 34 μονάδες. Για να περάσετε το μάθημα πρέπει να πάρετε τουλάχιστον 12 σε αυτό.

Στις ερωτήσεις πολλαπλής επιλογής πρέπει να επιλέξετε τη σωστή απάντηση (σε όλες τις περιπτώσεις, η σωστή απάντηση είναι μοναδική). Κάθε σωστή απάντηση δίνει 3 μονάδες. Κάθε λάθος απάντηση αφαιρεί 1 μονάδα. Αν ένα ερώτημα μείνει αναπάντητο δεν προσθέτει ούτε αφαιρεί μονάδες.

Στις ερωτήσεις που απαιτούν σύντομη απάντηση η βαθμολογική βαρύτητα είναι 3 μονάδες. Σε αυτές τις ερωτήσεις δεν υπάρχει αρνητική βαθμολογία.

Προγραμματιστικές Τεχνικές - Εξέταση "επί πτυχίω" - 25/2/2021 - Διάρκεια 90 λεπτά

3

Προγραμματιστικό θέμα.

Ορίστε τον τύπο του δυαδικού δέντρου tree με ακέραιες τιμές στους κόμβους.

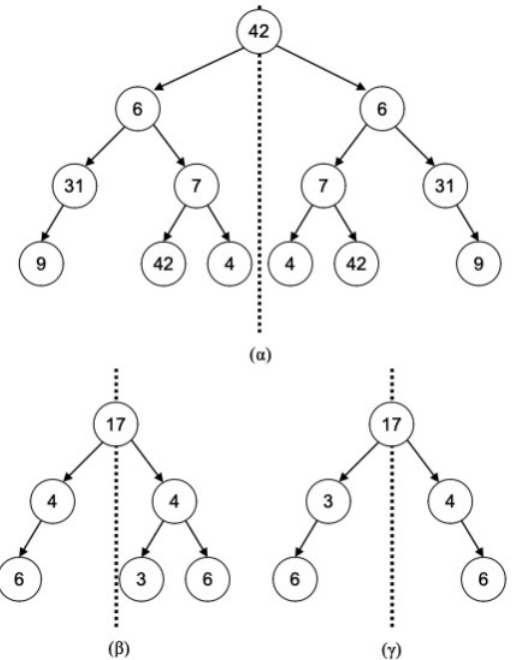
Θα ονομάζουμε ένα δέντρο «συμμετρικό» όταν έχει κατακόρυφο άξονα συμμετρίας, δηλαδή όταν είναι το ίδιο αν το δούμε σε καθρέπτη. Π.χ. το δέντρο (α) του διπλανού σχήματος είναι συμμετρικό, ενώ το δέντρο (β) δεν είναι γιατί δεν υπάρχει αντίστοιχος του κόμβου 3, ούτε το δέντρο (γ) γιατί ο αντίστοιχος του κόμβου 3 έχει την τιμή 4.

Γράψτε μια κομψή και αποδοτική συνάρτηση που να δέχεται ως παράμετρο ένα δένδρο και να ελέγχει αν είναι ή όχι συμμετρικό. Η επικεφαλίδα της συνάρτησης θα πρέπει να είναι η ακόλουθη:

```
bool symmetric(tree t);
```

Υπόδειξη: Πιθανώς θα χρειαστεί να γράψετε μια βοηθητική συνάρτηση.

(34 βαθμοί)

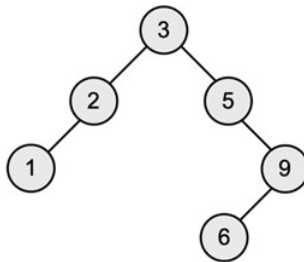


4

Δίνεται το δένδρο του παρακάτω σχήματος. Με βάση τους ορισμούς των δένδρων AVL ο κόμβος με το κλειδί 3 είναι:

(3 βαθμοί)

- ☐ (δεν απαντώ)
- ☐ Ίσα ψηλός (EH)
- ☐ Εκτός ισορροπίας (IMB)
- ☐ Δεξιά ψηλός (RH)
- ☐ Αριστερά ψηλός (LH)



5

Θεωρούμε το παρακάτω πρόγραμμα. Τι τυπώνεται κατά την εκτέλεσή του;

(3 βαθμοί)

```
class C {  
public:  
    void setX(int a) { x = a; }  
    C(int b): x(b) { y = b; }  
    void status() const { cout << x <<  
y; }  
private:  
    int x;  
    static int y;  
};
```

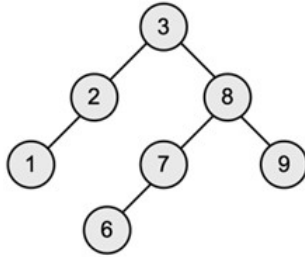
```
int C::y = 1;  
  
int main() {  
    C c1(7); c1.status(); c1.setX(4);  
    C c2(2); c2.setX(9); c2.status();  
    C c3(9); c1.status(); cout << endl;  
}
```

6

Δίνεται το δυαδικό δένδρο αναζήτησης του παρακάτω σχήματος, από το οποίο διαγράφεται ένα στοιχείο (δεν γνωρίζουμε ποιο). Κατά την εκτέλεση του γνωστού αλγόριθμου διαγραφής στοιχείων από δυαδικά δένδρα αναζήτησης, όποιο και να ήταν το στοιχείο που διαγράφηκε, τελικά θα γίνουν το πολύ:

(3 βαθμοί)

- ☐ (δεν απαντώ)
- ☐ 2 συγκρίσεις κλειδιών
- ☐ Καμία από τις άλλες απαντήσεις δεν είναι σωστή
- ☐ 4 συγκρίσεις κλειδιών
- ☐ 3 συγκρίσεις κλειδιών



7

Θεωρούμε το παρακάτω πρόγραμμα. Τι τυπώνεται κατά την εκτέλεσή του;

(3 βαθμοί)

```

int f(int x) {
    if (x < 10) return x;
    if (x < 20) throw
    true;
    return x-25;
}

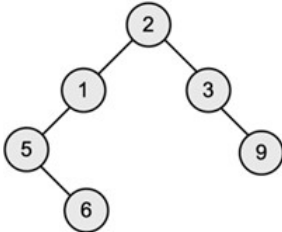
int main() {
    try { cout << f(42); cout << f(17); cout << f(7); }
    catch (bool b) { cout << 37; }
    catch (int n) { cout << n/6; }
    cout << endl;
}
  
```

8

Δίνεται το δένδρο του παρακάτω σχήματος. Ποιος από τους παρακάτω αλγόριθμους αναζήτησης θα επεξεργαστεί (επισκεφθεί) νωρίτερα τον κόμβο με το κλειδί 6;

(3 βαθμοί)

- ☐ Ο αλγόριθμος διάσχισης κατά πλάτος
- ☐ Ο αλγόριθμος προδιατεταγμένης (pre-order) διάσχισης κατά βάθος
- ☐ Ο αλγόριθμος μεταδιατεταγμένης (post-order) διάσχισης κατά βάθος
- ☐ (δεν απαντώ)
- ☐ Ο αλγόριθμος ενδοδιατεταγμένης (in-order) διάσχισης κατά βάθος

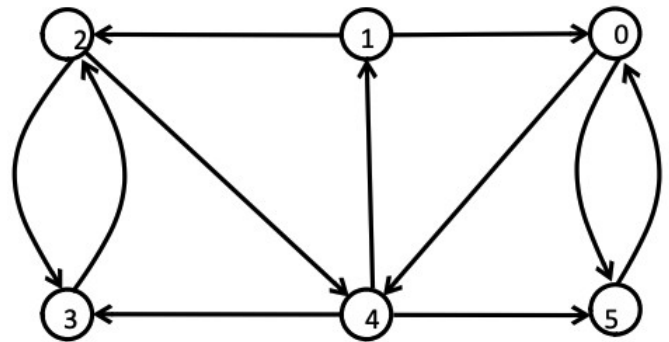


9

Εκτελέσετε τον αλγόριθμο εξερεύνησης κατά βάθος (DFS) στον γράφο του σχήματος με αφετηρία τον κόμβο $X = AM_last \bmod 6$, όπου AM_last = το τελευταίο ψηφίο του Αριθμού Μητρώου σας. Ποιος θα είναι ο 5ος κατά σειρά κόμβος που θα επισκεφθεί ο αλγόριθμος; Υποθέστε ότι ο 1ος κόμβος είναι η αφετηρία και ότι σε περίπτωση επιλογής γείτονα προηγείται πάντα ο μικρότερος κόμβος.

(3 βαθμοί)

- ☐ Ο κόμβος 0
- ☐ Ο κόμβος 4
- ☐ (δεν απαντώ)
- ☐ Ο κόμβος 2
- ☐ Καμία από τις άλλες απαντήσεις δεν είναι σωστή



10

Ποια είναι η έξοδος από την εκτέλεση του παρακάτω κώδικα C++;

(3 βαθμοί)

```

void swap1(int &i, int j) {
    int temp = i; i = j; j = temp;
}

void swap2(int i, int &j) {
    int temp = i; i = j; j = temp;
}

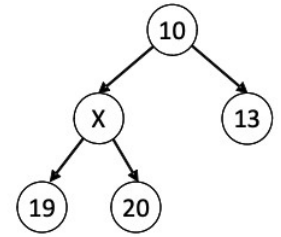
int main() {
    int a = 42, b = 17;
    cout << a << " " << b << " ";
    swap1(a, b);
    cout << a << " " << b << " ";
    swap2(a, b);
    cout << a << " " << b << endl;
}
  
```

11

Δίνεται ο εξής σωρός ελαχίστου. Θεωρήστε ότι αν $\text{last_AM} \leq 3$ τότε $X = 2 * \text{last_AM} + 11$, αν $3 < \text{last_AM} \leq 5$, τότε $X = 2 * \text{last_AM} + 5$ και αν $5 < \text{last_AM}$ τότε $X = 2 * \text{last_AM}$, όπου last_AM το τελευταίο ψηφίο του Αριθμού Μητρώου σας. Ποιες συγκρίσεις θα πραγματοποιηθούν κατά τη διαγραφή του στοιχείου 10;

(3 βαθμοί)

- ☐ Το 19 θα συγκριθεί πρώτα με το 10 και μετά με το X
- ☐ Το X θα συγκριθεί με το 13 και το 20 θα συγκριθεί με το X
- ☐ Καμία από τις άλλες απαντήσεις δεν είναι σωστή
- ☐ Το X θα συγκριθεί με το 13 και το 13 θα συγκριθεί με το 20
- ☐ (δεν απαντώ)
- ☐ Το 20 θα συγκριθεί πρώτα με το X και μετά με το 19



12

Θεωρούμε τον παρακάτω κώδικα. Έστω ότι ο πίνακας a έχει μέγεθος n . Ποια είναι η χρονική πολυπλοκότητα της συνάρτησης;

(3 βαθμοί)

```
int countPairDouble(std::vector<int> &a) {
    int n = a.size();
    std::sort(a.begin(), a.end());
    int result = 0;

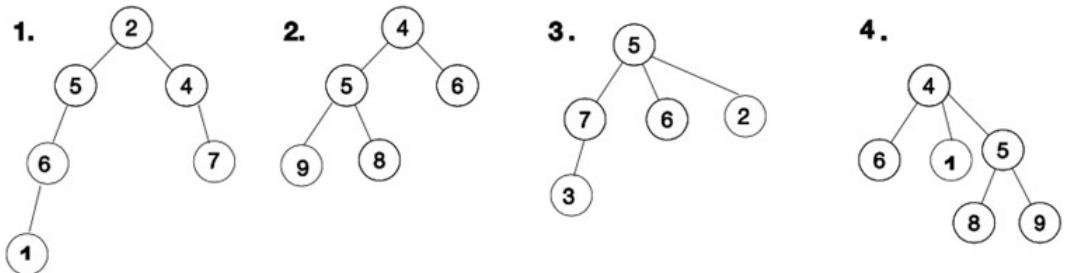
    for (int i = 0, j = 1; j < n; ++i) {
        while (j < n && a[j] < 2 * a[i]) ++j;
        if (a[j] == 2 * a[i]) ++result;
    }
    return result;
}
```

13

Δίνονται τα παρακάτω δένδρα που απεικονίζουν σύνολα μιας δομής Union-Find. Ποια από αυτά μπορεί να έχουν προκύψει με ακριβώς τέσσερις (4) εφαρμογές της διαδικασίας Union by Size αν θεωρήσουμε ότι τα αρχικά σύνολα ήταν όλα μονοσύνολα;

(3 βαθμοί)

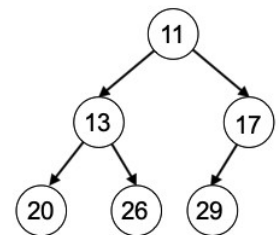
- ☐ Κανένα
- ☐ Το 1 και το 2
- ☐ (δεν απαντώ)
- ☐ Μόνο το 3
- ☐ Το 3 και το 4



14

Δίνεται ο εξής σωρός ελαχίστου. Σε αυτόν γίνεται εισαγωγή στοιχείου με τιμή $X = 3 * \text{AM_last}$, όπου AM_last το τελευταίο ψηφίο του Αριθμού Μητρώου σας, και στη συνέχεια εκτυπώνεται το δέντρο που προκύπτει με διάσχιση κατά πλάτος. Με ποια σειρά θα εκτυπωθούν οι επτά τιμές που θα περιέχει ο σωρός;

(3 βαθμοί)

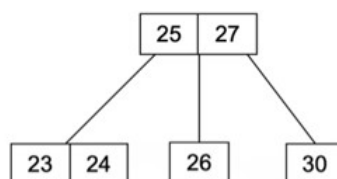


15

Δίνεται το B-δένδρο 3 οδεύσεων του παρακάτω σχήματος, στο οποίο αποθηκεύουμε ως κλειδιά φυσικούς αριθμούς. Για την εισαγωγή στοιχείων χρησιμοποιούμε το γνωστό αλγόριθμο εισαγωγής στοιχείων σε B-δένδρα. Ποιος είναι ο μέγιστος αριθμός στοιχείων που μπορούν να εισαχθούν σε αυτό το δέντρο, χωρίς να αλλάξει ύψος;

(3 βαθμοί)

- ☐ 1
- ☐ 0
- ☐ Καμία από τις άλλες απαντήσεις δεν είναι σωστή
- ☐ (δεν απαντώ)
- ☐ 2



16

Έχουμε ορίσει μία κλάση τα αντικείμενα της οποίας παριστάνουν διανύσματα και θέλουμε να υλοποιήσουμε την πράξη του εσωτερικού γινομένου, ορίζοντας κατάλληλα τον operator*.

(3 βαθμοί)

- ☐ Μπορούμε να τον ορίσουμε είτε ως μέθοδο της κλάσης μας με μία παράμετρο είτε ως φιλική (friend) συνάρτηση με δύο παραμέτρους
- ☐ Μπορούμε να τον ορίσουμε ως μέθοδο της κλάσης μας με μία παράμετρο
- ☐ (δεν απαντώ)
- ☐ Μπορούμε να τον ορίσουμε ως φιλική (friend) συνάρτηση με δύο παραμέτρους
- ☐ Μπορούμε να τον ορίσουμε ως μέθοδο της κλάσης μας με δύο παραμέτρους

17

Θεωρούμε το παρακάτω πρόγραμμα. Τι τυπώνεται κατά την εκτέλεσή του;

(3 βαθμοί)

```
class one {
public:
    virtual void foo() { cout << "A"; bar(); }
    void bar() { cout << "B"; }
};

class two : public one {
public:
    void foo() override { cout << "C";
bar(); }
};

class three : public one {
public:
    void bar() { cout << "D"; }
};

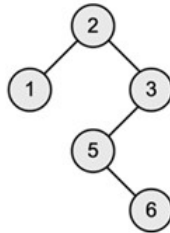
int main() {
    one *p;
    p = new one;    p->foo();
    p = new two;    p->foo();
    p = new three;  p->foo();
    cout << endl;
}
```

18

Δίνεται το δένδρο του παρακάτω σχήματος. Πόσα διαφορετικά μονοπάτια έχει αυτό το δένδρο;

(3 βαθμοί)

- ☐ Καμία από τις άλλες απαντήσεις δεν είναι σωστή
- ☐ 7
- ☐ 4
- ☐ 12
- ☐ (δεν απαντώ)



19

Τι θα συμβεί αν ξεχάσουμε τη λέξη "virtual" στην επικεφαλίδα του καταστροφέα μιας βασικής κλάσης;

(3 βαθμοί)

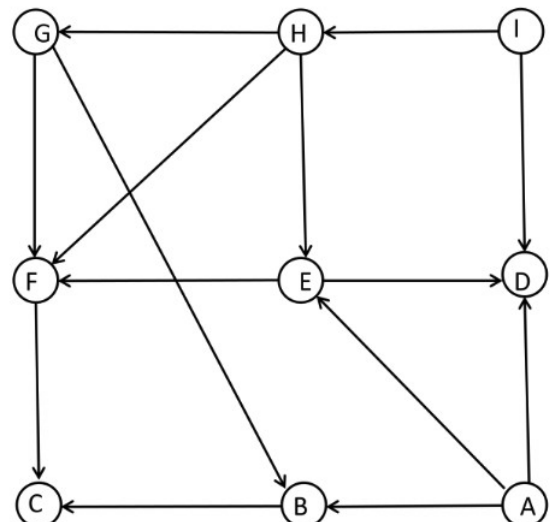
- ☐ Το πρόγραμμα θα βγάλει σφάλμα κατά τη μεταγλώττιση
- ☐ Δε θα καταστρέφονται πάντοτε σωστά τα αντικείμενα παραγόμενων κλάσεων
- ☐ Τίποτα, όλα καλά...
- ☐ Δε θα μπορούμε να ορίσουμε παραγόμενες κλάσεις
- ☐ (δεν απαντώ)
- ☐ Δε θα καταστρέφονται πάντοτε σωστά τα αντικείμενα της βασικής κλάσης

20

Ποια από τις παρακάτω ακολουθίες είναι τοπολογική ταξινόμηση για το γράφο του σχήματος;

(3 βαθμοί)

- ☐ A-I-H-E-G-B-D-F-C
- ☐ H-I-G-D-E-F-C-A-B
- ☐ (δεν απαντώ)
- ☐ Καμία από τις άλλες απαντήσεις δεν είναι σωστή
- ☐ I-H-D-G-E-F-A-B-C
- ☐ Περισσότερες από μία από τις άλλες απαντήσεις είναι σωστές
- ☐ A-I-H-F-G-B-D-E-C

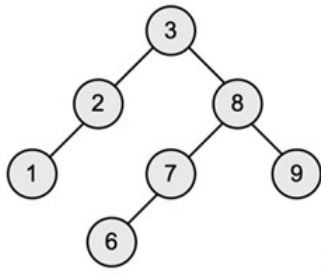


21

Στο δένδρο AVL του παρακάτω σχήματος εισάγεται το κλειδί 4 και στη συνέχεια διαγράφεται το κλειδί 6. Ποια από τις επόμενες προτάσεις που αφορούν το τελικό δένδρο AVL είναι σωστή;

(3 βαθμοί)

- ☐ Ο κόμβος με το κλειδί 3 είναι ίσα ψηλός (EH)
- ☐ Ο κόμβος με το κλειδί 4 είναι ίσα ψηλός (EH)
- ☐ Καμία από τις άλλες απαντήσεις δεν είναι σωστή
- ☐ Ο κόμβος με το κλειδί 4 είναι δεξιά ψηλός (RH)
- ☐ (δεν απαντώ)

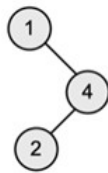


22

Δίνεται το παρακάτω δυαδικό δένδρο αναζήτησης. Χρησιμοποιώντας το γνωστό αλγόριθμο εισαγωγής στοιχείων σε δυαδικά δένδρα αναζήτησης, με πόσες διαφορετικές ακολουθίες εντολών εισαγωγής στοιχείων θα μπορούσε να έχει σχηματιστεί το δένδρο αυτό, θεωρώντας ότι κάθε στοιχείο εισάγεται μόνο μία φορά;

(3 βαθμοί)

- ☐ 1
- ☐ 2
- ☐ (δεν απαντώ)
- ☐ Δεν μπορεί να σχηματιστεί με κανέναν τρόπο
- ☐ 4



23

Δίνονται τα κλειδιά 1,2,3,4. Πόσα διαφορετικά AVL δένδρα αναζήτησης μπορούν να περιέχουν ακριβώς αυτά τα τέσσερα κλειδιά σε κόμβους τους;

(3 βαθμοί)

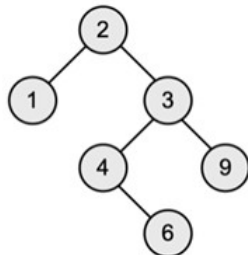
- ☐ (δεν απαντώ)
- ☐ 2
- ☐ Καμία από τις άλλες απαντήσεις δεν είναι σωστή
- ☐ 3
- ☐ 4

24

Δίνεται το δένδρο του παρακάτω σχήματος. Ποιος είναι το μέγιστο μήκος στο οποίο θα φτάσει η ουρά που χρησιμοποιείται από τον αλγόριθμο διάσχισης κατά πλάτος, έως ότου ολοκληρώσει τη διάσχιση αυτού του δένδρου;

(3 βαθμοί)

- ☐ 2
- ☐ Καμία από τις άλλες απαντήσεις δεν είναι σωστή
- ☐ (δεν απαντώ)
- ☐ 1
- ☐ 3



25

Δίνεται το δένδρο του παρακάτω σχήματος. Πόσα μη κενά υποδένδρα του είναι δυαδικά δένδρα αναζήτησης;

(3 βαθμοί)

- ☐ Καμία από τις άλλες απαντήσεις δεν είναι σωστή
- ☐ (δεν απαντώ)
- ☐ 3
- ☐ 7
- ☐ 6

